

FEB 11 2005 14:56 FR FULBRIGHT&JAWORSKI 212 318 3111 TO 0193310001987170 P.79

EXHIBIT G

- 1 -

Document3

PAGE 79/90 * RCVD AT 2/11/2005 2:40:25 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/4 * DNIS:8729306 * CSID:2123183111 * DURATION (mm:ss):20:50

FEB 11 2005 14:56 FR FULBRIGHT&JAWORSKI 212 318 3111 TO 0193310001987170 P.80
Architectu. Design of InterScope

Architecture Design of InterScope™

Document ID :
Owner :

Virtual Gold Confidential

Page 1 of 1

Virtual Gold Confidential

PAGE 80/90 * RCVD AT 2/11/2005 2:40:25 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/4 * DNIS:8729306 * CSID:212 318 3111 * DURATION (mm:ss):20:50

1. Introduction

InterScope is a tool aimed at web browser users to enable them to mine tabular data on an Intranet / Internet. It would enable them to discover interesting patterns in the data, with minimal inputs.

InterScope™ aims at fulfilling the following requirements :

- (a) Basic table functionality on the web – the user should be able to represent the data from any data source using the InterScope™ ActiveTable™. The table should have the same look and feel as a normal HTML table.
- (b) FastPatterns – when the user moves her mouse on a column of the ActiveTable™, she should be able to view some interesting facts about the column.
- (c) Data Mining, using secondary data sources – on the click of a button, the user should be able to mine data from several data sources, and link in information from other data sources (such as video)
- (d) User Question Support - At the click of a button, the user should be able to frame and ask her own question, and get answers from different data sources, also linking her to secondary sources of information (such as text files, or video clips)

Section 2 describes the InterScope™ design goals. Section 3 describes the InterScope™ architecture.

2. Design Goals

The InterScope™ design is aimed at satisfying the following constraints :

2.1. Customizability

Reusability of InterScope™ components was a primary design goal. An InterScope™ table will be designed by a web developer, who should be able to mix and-match various InterScope™ components together to create a solution that best fits her needs. For example, a developer might decide to include only basic ActiveTable™ functionality on her web page, without any data mining / user question support. On the other hand, she could choose to include FastPattern, data mining and User Question support along with the basic ActiveTable™. She should be able to build solutions to meet both these scenarios, as well as various options in between, using standard Builder tools.

2.2. Modularity

A primary design goal was to provide support for the requirements mentioned in Section 1 in a phased manner. The requirements are to be addressed in the

order given above. Hence, the design should be modular to allow for easy addition of new features.

2.3. Flexibility

InterScope™ is to be used on a wide variety of platforms on Intranets / the Internet. At the time of design, it is not possible to predict the nature of the end-user environment. The end-user could be working on platforms ranging from embedded devices (such as a pager) and "thin" clients or network computers, to powerful desktop machines and workstations. This environment could have major effects on the performance of the InterScope™ table, especially when computation-intensive data mining queries are being run.

The design should be flexible, such that the ActiveTable™ is being designed by the Internet / Intranet developer, portions (or all) of the data and code can reside on the web server, and be brought to the client, as and when required. The design should support this distributed functionality.

3. InterScope™ Architecture

The main component of InterScope™ is the ActiveTable™. This is an "active" representation of tabular data, in which each tabular element can be made to respond to user actions (such as mouse movements).

3.1. The ActiveTable™

The ActiveTable™ is the basic InterScope™ component. It is always run on the client.

3.1.1. ActiveTable™ Properties

An ActiveTable™ has various properties, such as the Data Source, text color, font size, etc., all of which the web developer can specify while designing the ActiveTable™. (The developer will be assisted by an ActiveTable™ customizer wizard while designing the ActiveTable™). Except for the Data Source property, all of the others will have standard default values.

3.1.2. ActiveTable™ Elements

An ActiveTable™ has the following elements :

3.1.2.1 TitleGridArray

A single-row GridArray, consisting of a number of GridElements, for representing the titles of the various columns. This is always sent to the client. The title text properties of the ActiveTable™ are reflected by the TitleGridArray

3.1.2.2 DataGridLayout

A GridLayout representing the actual data. This is always sent to the client. The normal text properties of the ActiveTable™ are reflected by the GridLayout. In the GridLayout may have scroll bars to aid the user in viewing large tables. In addition, it has an array of ColumnManager elements, each of which contains :

- (1) Information about the totals of various "singles" in a column.
- (2) Information relating to the categorization of the column

3.2. The ClusterSet

The developer can add one or more Clusters to a basic ActiveTable™ to induce FastPattern and Data Mining functionality. A cluster is thus a JavaBean, that the user can add to an ActiveTable™. These clusters are stored in a ClusterSet. (As in the case of the basic ActiveTable™, the developer would be assisted by a wizard when she designs clusters).

Each cluster will contain the following information :

- (1) Information about the attributes in the cluster
- (2) Information about the "singles" in the cluster
- (3) FastPattern information about the cluster

The ClusterSet could exist only on the server or on both the client and the server, depending on the size of the ActiveTable™ data. If the data are small, the ClusterSet is stored on the server, as well as attached to the ActiveTable™ applet that would be sent to the client at runtime. If the table is large, it is stored only on the server. This decision is (automatically) taken at the time the ActiveTable™ is designed, transparent to the developer, according to some criterion, still to be developed.

In either case, code is added to the ActiveTable™ to allow for remote invocation of the ClusterSet's methods.

3.2.1. ClusterSet Component Elements

3.2.1.1 FastPatternBalloon

The FastPatternBalloon component is added automatically to the ActiveTable™ when the developer adds a cluster to the ActiveTable™. This is activated when the user's mouse is on a column of the ActiveTable™ that belongs to a cluster that the developer has defined, and displays a FastPattern (some interesting information relating the column to other attributes in the cluster).

The FastPattern balloon is always sent to the client along with the ActiveTable™.

3.3. The DataMiner Component

The developer can add a DataMiner Component to the ActiveTable™ to enable data mining across different data sources. The DataMiner component customizer wizard would collect information about the secondary data sources, etc. from the developer. At least one cluster should have been created before the DataMiner component can be added, as the mining is performed on the attributes in a cluster. (If a cluster has not been created, the developer will be prompted to create one, by the DataMiner customizer).

The DataMiner component contains the code (Attribute Focusing) for data mining. It is treated similar to the ClusterSet (retained only on the server for large tables, and kept on both client and server for small data sources).

3.4. The YourQuestion Component

The developer can add this component to an ActiveTable™ to enable a user to ask her own questions of the data. If other data sources have been added, the user will be able to ask questions across different sources. It is treated similar to the ClusterSet (retained only on the server for large tables, and kept on both client and server for small data sources).

3.4.1. YourQuestion Component Elements

3.4.1.1 Index

The index is a means of fast access to the data in the DataGridLayout and aids in answering user questions. It is added to the ActiveTable™ when the YourQuestion component is added to it.

3.5. Design Scenarios

The following table depicts various possibilities for mixing and matching the InterScope™ components :

Configuration	Functionality
ActiveTable™	Basic table functionality
ActiveTable™ + ClusterSet	Basic table functionality + FastPatterns
ActiveTable™ + ClusterSet + DataMiner	Basic table functionality + FastPatterns + Data Mining capability across data sources
ActiveTable™ + ClusterSet + YourQuestion	Basic table functionality + FastPatterns + User question capability across data sources
ActiveTable™ + ClusterSet + DataMiner + YourQuestion	Basic table functionality + FastPatterns + Data Mining capability across data sources + User question capability across data sources